Real-Time Taxi–Passenger Matching Using a Differential Evolutionary Fuzzy Controller

Yue-Jiao Gong, *Member, IEEE*, Yi-Wen Liu, Ying Lin, *Member, IEEE*, Wei-Neng Chen, *Senior Member, IEEE*, and Jun Zhang, *Fellow, IEEE*

***Abstract*—Real-time taxi–passenger matching plays a critical role in modern taxi dispatch systems. Currently, the greedy strategy is widely adopted, which limits the quality of the service (QoS) and the profit of the entire system. There are two crucial tasks in this system: 1) the pairwise prioritization and 2) the matching of taxi–passenger pairs. In this paper, we develop a two-stage taxi–passenger matching system to deal with these two tasks. In the first stage, we design a fuzzy controller to assign a priority score to each taxi–passenger pair in real time. To ensure its performance on providing good QoS and profit, the fuzzy controller is optimized by an offline differential evolution algorithm. New individual representation is designed to optimize the membership functions and fuzzy rule base simultaneously. To accelerate the optimization process, the algorithm is implemented in a parallel way. Then, in the second stage, considering the priority scores as weights in the bipartite graph of taxi and passenger sets, we further apply a polynomial Kuhn–Munkres algorithm to find the maximum weight perfect matching in the bipartite graph. Simulated results validate the effectiveness of the proposed algorithm, which is able to enhance the QoS provided by the taxi system and improve the profit gained by the taxi service company.**

***Index Terms*—Differential evolution (DE), fuzzy logic controller (FLC), Kuhn–Munkres (KM) algorithm, taxi dispatch system.**

I. Introduction

THE TAXI–PASSENGER matching problem plays a crucial role in modern taxi dispatch systems, since it is closely related to the experience of the passengers and the profit of the taxi service provider. However, the traditional taxi dispatch is lack of an efficient method to match the passengers and the available taxis. Commonly, the drivers roam around the city to pick up the passengers on the roadside. The drivers would like to hunt for the same passengers in some hotspot areas, which may aggravate the traffic congestion problem [1]–[3]. Meanwhile, the passengers in some other places can hardly get served, which lowers the quality of service provided by the taxi system. Considering these deficiencies, an efficient taxi dispatch method is in need to improve the satisfaction of the users as well as to increase the profit of the company.

With the proliferation of mobile taxi booking applications, the application server receives the real-time information of taxis and passengers and decides the taxi–passenger assignment. However, the current taxi dispatch systems usually adopt a greedy strategy to match the taxis and passengers. The nearest neighbor (NN) algorithm and the max profit (MP) algorithm combined with a first come first serve (FCFS) strategy are commonly used, since they are easy to implement and manage [4]–[6]. Although the NN with FCFS method can guarantee the priority of the passengers and reduce the waiting time of passengers, it handles the matching process in a local perspective. The method matches passengers and taxis considering only the distance between passengers and taxis, whose comprehensive performance is unsatisfactory. The MP with FCFS method targets at gaining maximum profit in each single match, but it has two limitations. First, the method does not consider the passenger's experience, so that the waiting time of the passengers is often much longer than that derived by NN. Second, the method always increases the traverse distance of a taxi to pick up a passenger.

To further improve the taxi dispatch systems, research efforts are paid to developing new effective matching algorithms. Gao *et al.* [7] proposed a new algorithm called KMBA, which combines the Kuhn–Munkres (KM) algorithm with NN to match passengers and taxis in each time window. Meghjani and Marczuk [8] developed a hybrid method that uses the distance-based travel cost to fast pair the taxis and passengers. Maciejewski *et al.* [9] proposed an assignment approach that takes a more global look in making dispatch decisions. The experiments confirm that it can reduce the average waiting time and enhance the utility of the taxis. Zheng *et al.* [10] proposed a greedy algorithm with a packing-based matching strategy (GPBM) to maximize the profit of taxi–passenger matching. Miao *et al.* [11] proposed a receding horizon control approach that considers the predicted demands

Manuscript received October 6, 2018; revised February 1, 2019; accepted May 7, 2019. Date of publication May 31, 2019; date of current version April 15, 2021. This work was supported in part by the National Natural Science Foundation of China under Grant 61873095, Grant 61772569, Grant 61873097, and Grant U1701267, and in part by the Science and Technology Planning Project of Guangzhou under Grant 201904010211. This paper was recommended by Associate Editor H. Ying. *(Corresponding authors: Ying Lin; Jun Zhang.)*

Y.-J. Gong, Y.-W. Liu, and W.-N. Chen are with the School of Computer Science and Engineering, South China University of Technology, Guangzhou 510006, China, and also with the Guangdong Provincial Key Laboratory of Computational Intelligence and Cyberspace Information, South China University of Technology, Guangzhou 510006, China.

Y. Lin is with the Department of Psychology, Sun Yat-sen University, Guangzhou 510006, China (e-mail: linying23@sysu.edu.cn).

J. Zhang is with the Institute of Sustainable Industries and Liveable Cities, University of Victoria, Melbourne, VIC 8001, Australia (e-mail: junzhang@ieee.org).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TSMC.2019.2916184.

Digital Object Identifier 10.1109/TSMC.2019.2916184

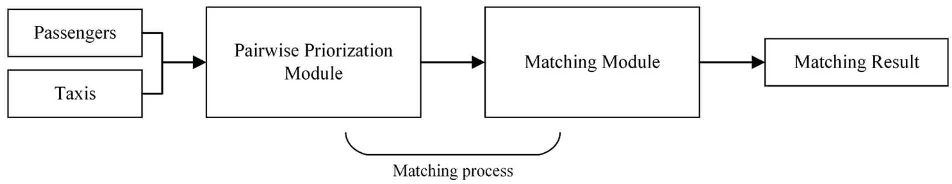


Fig. 1. Pipeline of the proposed taxi dispatch system.

to control taxis. In [12], a multiagent taxi dispatch system had been introduced, which adopted a negotiation strategy to help drivers find an optimal assignment solution that they both agree. This method can reduce the waiting time of passengers and also satisfy the requirements of drivers. In [13] and [14], a fuzzy logic controller (FLC) considering two contradictory rules was applied to the dispatch system. The results showed that the method helps to keep a uniform workload of the system while its performance in service time is close to that of the NN. In [15], a fuzzy taxi scheduling system (FTSS) was proposed, which incorporates five criteria to make decisions for the taxi scheduling. The experiments also validate the advantages of the FLC. In addition, some efforts have been paid to developing route recommendation systems for drivers to hunt for passengers on streets [16]–[18].

Overall, greedy strategies are still the mainstream method for matching taxis and passengers. But they have some shortcomings. Most of these greedy strategies derive the taxi–passenger matching degree from a single attribute, such as the taxi–passenger distance or profit of the deal. The taxi–passenger pairs with high matching degrees are then favored in the following matching procedure. Despite of the advantages of easy implementation and generality, the accuracy of the greedy results is insufficient, because the matching degree derived from a single attribute usually fail to reflect the overall status of the taxi–passenger pair. The passengers that score low in the specific attribute may have to wait a long time or even cancel their requests, which reduces both the quality of the service (QoS) and the profit of the taxi dispatch system. For obtaining optimal results in both terms of QoS and profit, the taxi dispatch system must take multiple matching attributes into account.

In this paper, we develop a two-stage taxi–passenger matching system, which is illustrated in Fig. 1. The first stage calculates the matching degree for each possible taxi–passenger assignment, which is named as the pairwise prioritization module. After that the second stage is the matching module to solve the optimal assignment problem, which will provide the final matching result. The objective of the taxi dispatch system is to maximize the QoS and increase the profits for the taxi service company. As reviewed above, the traditional dispatch system is limited in considering only the distance or the profit of passengers. In contrast, in our pairwise prioritization module, we include more attributes to let our system be more comprehensive to cope with different situations. A simple method is to linearly accumulate the normalized input attributes to an aggregation function. However, the linear aggregation is somewhat trivial and limited in the performance. The FLC can incorporate multiple input attributes and simulate the intelligent decision making procedure of human by performing several linguistic rules [19]–[21]. In addition, FLC is stable and computationally efficient, which can provide real-time responses to the system [14], [22], [23]. In this paper, the FLC processes three input attributes: 1) the waiting time of the passenger; 2) the travel distance of the passenger; and 3) the distance between the passenger and the taxi, while it outputs a priority score. As for the matching module, traditionally, the FCFS is widely adopted, whose performance is unsatisfactory since it matches the passengers in a very local perspective. In our system, based on the output of FLC, we construct a weighted bipartite graph for taxis and passengers. This graph is the input of the matching module. A global algorithm, the KM algorithm, is then adopted to achieve the maximum weight perfect matching.

In the traditional FLC, the membership functions and the fuzzy rule base could be chosen by experts. However, without sufficient experience, one can hardly design an optimal FLC that can maximize the performance of the taxi dispatch system. Evolutionary algorithms (EAs) do work in such a scenario, since they can gain experience during the population-based search process generation by generation. In the related area, many efforts have been paid to adopting EAs to optimize the membership functions. The experimental results validated the effectiveness in enhancing the performance of FLC [24]–[30]. In addition, some other researches focused on the optimization of the fuzzy rule base, which also showed significant improvement of the FLC [31]–[33]. However, the optimization of a single component of FLC is insufficient, since the membership functions and fuzzy rule base both play crucial roles in the FLC. The optimization of the combination of the membership functions and fuzzy rule base is a mix-domain optimization problem, which is a challenging issue. To deal with this challenge, we develop a differential evolution (DE) algorithm to optimize the membership functions and fuzzy rule base simultaneously. DE is one of the most competitive population-based stochastic optimization algorithms in dealing with NP-hard problems [34]–[41]. In the proposed algorithm, a new individual structure is put forward to represent the membership functions. Then, each individual is companied with a medium-state rule table, which is determined automatically by an association scheme we developed. Through the association method, the membership functions and fuzzy rule base are optimized simultaneously in the evolution process. In addition, to accelerate the optimization process, the DE is implemented in a parallel way under the environment of the message passing interface (MPI). Note that we train (optimize) the FLC settings offline and then perform the FLC online for realizing the real-time taxi–passenger matching (RTTPM).

To summarize, the contributions of this paper are presented as follows.

1) We propose a novel optimization model to solve the RTTPM problem in a more comprehensive manner such that the QoS and the profit of the taxi dispatch system can be optimized simultaneously. To achieve efficient optimization, three critical attributes (i.e., the waiting time of the passenger, the travel distance of the passenger, and the distance between the passenger and taxi) are extracted from the taxi hailing process and considered in the model.
2) We design a two-stage architecture for the taxi dispatch system. As shown in Fig. 1, the two-stage system consists of a pairwise prioritization module and a matching module. The prioritization module uses an FLC to compute the matching degree of each taxi–passenger pair based on the above three attributes. The matching model converts the problem into a bipartite graph weighted by matching degrees and employs the KM algorithm to find the perfect matching. To the best of our knowledge, we are the first attempt to conduct prioritization before matching taxis with passengers, which enables the system to handle complex practical situations.
3) We develop a novel parallel DE (PDE) algorithm to optimize the FLC. The PDE uses an association-based representation scheme that allows concurrent optimization of the membership functions and the fuzzy rule base of the FLC, while traditional FLC optimization methods only consider one of the two components. The fitness function of PDE is set as the simulation result of the taxi dispatch system, which facilitates direct optimization of the FLC for achieving excellent system performance. Further, we implement the PDE on a parallel model to accelerate its execution speed. We name the resulting FLC as PDE-FLC.

Several simulations are conducted to validate the effectiveness and flexibility of the proposed taxi dispatch system. Simulation results show that it provides better QoS and profit than the recently proposed algorithms. Further, we conduct investigation simulations to evaluate the effectiveness coming from different subcomponents of the proposed system. It has been shown that concurrent optimization of the membership functions and the fuzzy rule base is useful for performance enhancement. Meanwhile, adopting KM in the matching module is much better than the FCFS.

The rest of this paper is organized as follows. Section II reviews related works. Section III models the RTTPM problem. In Section IV, the FLC-based taxi dispatch system is introduced in details. Section V illustrates the PDE-FLC. The simulations are presented in Section VI. Finally, the conclusions are drawn in Section VII.

II. Related Work

A. Related Taxi–Passenger Matching Methods

There are a number of RTTPM algorithms in the literature, which can be divided into three categories: 1) basic greedy algorithms; 2) predictive mechanisms involved greedy algorithms; and 3) fuzzy controller-based algorithms.

1) The basic greedy algorithms are mainly based on the NN and MP strategies [7]–[10], [12]. Namely, the smaller the distance between a passenger and a taxi or the higher profit the passenger can provide, the higher priority the passenger possesses. These algorithms are easy to implement and manage. In addition, some other adaption methods are attached to improve the performance. In [7], the KM algorithm was used to achieve the optimal matching. Besides, a negotiation strategy was proposed to improve the matching process in [12]. Although these methods help to enhance the performance of the greedy strategy, the limitations we discussed in the introduction still exist.
2) The predictive mechanisms involved algorithms utilize the historical data of the passengers and taxis to predict some future indices in the system [11], [42]–[44]. For example, the work in [42] predicts the future availability of a taxi which is currently occupied by a passenger. The study in [43] predicts the service demand that will emerge at a given taxi stand. The predictive mechanisms mine the historical data to gain more information about the system for possible performance enhancement. However, the related algorithms still perform the matching procedure in a greedy manner, which are deficient.
3) Owing to the great success of FLC in the industry, some researches attempted to use this method to the taxi–passenger matching. In [13], the experimental results showed that an FLC achieved similar performance of NN while it balanced taxi utilization. In [14], it was proved that this method can generally outperform the conventional methods that using a single criterion, such as min distance or max profit. Therefore, the FLC-based algorithms can be very competitive in the taxi–passenger matching. However, the research progress in this area is still limited and many crucial issues are remained unexplored. Different from [13] and [14], this paper develops a two-stage taxi–passenger matching system, which applies FLC to accomplish pairwise prioritization and then utilizes the KM algorithm to achieve optimal matching results. Besides, the critical components of FLC, such as the membership functions and fuzzy rule base, are optimized by the PDE algorithm. The optimization objective is to improve both QoS and system profit.

B. Differences Between Taxi–Passenger Matching and Route Recommendation

In addition, some route recommendation systems have also been proposed to improve the QoS and profit of the taxi service company. In [16] and [17], a coarse-grained recommendation system was proposed to provide promising driving directions for the taxi drivers. Qian *et al.* [18] proposed a sharing considered route assignment mechanism (SCRAM), which can provide recommendation fairness for a group of competing

TABLE I
NOTATIONS OF THE RTTPM

Parameters	Description
$G = (V, E)$	Taxi-passenger network
T	The total number of the time frames
ϕ_t	Passenger demand queue at time t
ψ_t	Available taxi queue at time t
M_t	The proximity matrix of matching degree at time t
H_t	The incident matrix of matching result at time t
F_t	Total matching degree at time t
Q_{total}	The total number of passengers
Q_{match}	The total number of matched passengers
D_i	The travel distance of passenger i
$E(i, j)$	Minimal distance between the passenger i and taxi j
$w_{i,j}$	Traverse time from the current location of taxi j to the location of passenger i.
b_i	The booking time of passenger i
α	Revenue per kilometer
β	Cost per kilometer
SR	Service rate
TW	The average waiting time of passenger
PR	Total profit

taxi drivers. It considers more variables to enhance the fairness and efficiency of the systems.

The route recommendation system mines the data of the taxi traces and the historical information of the passengers and provides promising driving routes for drivers, so that the drivers do not need to roam on the roads and hunt for passengers by themselves. However, the taxi–passenger matching system is different from the route recommendation system. The application server can collect the real-time information of the passengers and taxis. The matching system is designed to solve the real-time matching of the passengers and taxis. The taxi drivers only need to receive the matching result of the application server and then go to pick up his passengers. In this way, the drivers also do not need to roam on the road. Both of the two types of systems are quite valuable for the transportation fields. In this paper, we mainly focus on the design of the real-time taxi–passenger matching system.

III. Problem Formulation

A. Real-Time Taxi–Passenger Matching

This section presents the network and the necessary assumptions of the RTTPM problem. Table I shows the notations in the RTTPM. The taxi–passenger network is represented as an undirected graph $G = (V, E)$, where the vertex set V represents the available locations for passengers to get on or off the taxi and the edge set E is a routine set between locations in the network. We discretize the continuous time into many time frames $\{1, 2, \ldots, t, \ldots, T\}$. The parameter T is determined by the sampling time between two matching procedures. During the sampling time, the information about potential passengers and available taxis are collected to build the passenger queue and the taxi queue. It is based on the two queues that the dispatch system can consider the assignments of taxis to passengers from a global point of view, rather than using immediate information as in the FCFS methods. The setting of the sampling time is therefore crucial for an efficient dispatch system. If the sampling time is too short, the obtained information is very limited, and the system will degenerate to a local method using immediate information. If the sampling time is too long, both passengers and taxi drivers have to wait a long time before getting scheduled. In [10], it is suggested that the sampling time should be 2–20 s. In order to be as real time as possible, in this paper, we set the sampling time as 2 s. This setting is also used in the DiDi app, which is a widely used real-time taxi hailing app (like Uber) in China. Note that since the sampling time is introduced to perform matching, the working scenario is sometimes considered as "near-real time." Using a 2 s sampling time, the number of the time frames, T, is 1800/h.

Let ϕ_t and ψ_t denote the passenger demand queue and the available taxi queue at the time frame t, respectively. The matching algorithm is conducted to assign the $|\phi_t|$ passengers to the $|\psi_t|$ taxis.

More formally, given

$$\phi_t = \{p_1, p_2, \ldots, p_m\} \text{ and } \psi_t = \{v_1, v_2, \ldots, v_n\} \tag{1}$$

$$M_t[i, j] = f(p_i, v_j) \tag{2}$$

where $m = |\phi_t|$ and $n = |\psi_t|$ represent the number of the passengers and the taxis at time t. In addition, a numerical score named matching degree is used to describe the desirability to dispatch taxi i to serve passenger j. It is calculated in the pairwise prioritization module in the taxi dispatch system. For example, the previous studies directly adopted the distance between the passengers and taxis [7], [8] or the price of the deal [10] as the matching degree. So let f represent the matching degree function between a pair of taxi and passenger and M_t is an m by n proximity matrix denoting the matching degree at time t. Namely, $M_t[i, j]$ denotes the matching degree of assigning taxi v_j to the passenger p_i. In the proposed system, f is modeled in the pairwise prioritization module using FLC. Next, suppose the function h (p_i, v_j) represents the paring results of the passengers and taxis

$$h(p_i, v_j) = \begin{cases} 1, & \text{if taxi } v_j \text{ is assigned to passenger } p_i \\ 0, & \text{otherwise} \end{cases} \tag{3}$$

and H_t is the m by n incident matrix at time t as

$$H_t[i, j] = h(p_i, v_j) \tag{4}$$

subject to

$$\sum_{i=1}^{m} H_t[i, j] \leq 1, \sum_{j=1}^{n} H_t[i, j] \leq 1. \tag{5}$$

Namely, each passenger can only be served by one taxi, while each taxi can only serve one passenger at a time. The total matching degree F at time t is formulated as

$$F_t = \sum_{i=1}^{m} \sum_{j=1}^{n} M_t[i, j] H_t[i, j]. \tag{6}$$

The passenger who is not assigned a taxi at the time frame t will be passed to the next time frame to process. The system performance is evaluated by considering the matching results of all time frames, which will be detailed in the next section.

Besides, in real-life situation, the passenger could not endure too long waiting time for the service. So, it is assumed that the passengers will cancel their demands if they have waited for more than 300 time frames (10 min) in the RTTPM.

B. System Objectives

In the proposed system, the objective is to maximize the QoS and profit. The QoS is defined based on two measures: 1) the service rate and 2) the average waiting time of the passengers before they are picked up. In the following sections, we define these measures one by one.

1) Service Rate: The service rate is computed as

$$SR = \frac{Q_{\text{match}}}{Q_{\text{total}}} \tag{7}$$

$$Q_{\text{match}} = \sum_{t=1}^{T} \sum_{i \in \phi_t} \sum_{j \in \psi_t} H_t[i, j] \tag{8}$$

where Q_{match} denotes the quantity of passengers that have been matched, Q_{total} is the total number of passengers, and T is the number of the time frames. SR evaluates how much pressure the taxi dispatch system can support. The more passengers that the system can serve, the higher the SR can be, which is desired.

2) Waiting Time: The waiting time of passengers is caused by two issues. First is the time to wait the system's response for matching, which is denoted as T_{match}. The other is the waiting time for the taxi to arrive, which is represented as T_{pick}. The average time of waiting is formulated as

$$TW = T_{\text{match}} + T_{\text{pick}} \tag{9}$$

$$T_{\text{match}} = \frac{\sum_{t=1}^{T} \sum_{i \in \phi_t} \sum_{j \in \psi_t} H_t[i, j] \cdot (t - b_i + s_t)}{Q_{\text{match}}} \tag{10}$$

$$T_{\text{pick}} = \frac{\sum_{t=1}^{T} \sum_{i \in \phi_t} \sum_{j \in \psi_t} H_t[i, j] \cdot w_{i,j}}{Q_{\text{match}}} \tag{11}$$

where b_i is the booking time of the passenger i; s_t is the system runtime at time t; and $w_{i,j} = E(i, j)/\text{vel}$ is the traverse time from the current location of taxi j to the location of passenger i. The vel is the average velocity of the taxis. The TW is deeply related to the QoS. The low TW means that the passengers can be served in a short time, which improves the passengers' experience. Besides, all taxis are assumed to comply with an identical average velocity (35 km/h according to a recent report released by the Beijing Municipal Commission of Transport), so that the term "traverse distance" and "traverse time" can be used interchangeably.

3) Profit: The profit is a crucial criterion that the taxi service provider cares a lot. Usually, it is calculated by subtracting the expenses from the total income. In the real-life situation, the composition of the expenses could be very complicated. Nevertheless, the major expense is highly related to the traverse distance of taxis. In the simplified model, the profit PR is represented as

$$PR = \text{Income} - \text{Expense} \tag{12}$$

$$\text{Income} = \alpha \cdot \sum_{t}^{T} \sum_{i \in \phi_t} \sum_{j \in \psi_t} H_t[i, j] \cdot D_i \tag{13}$$

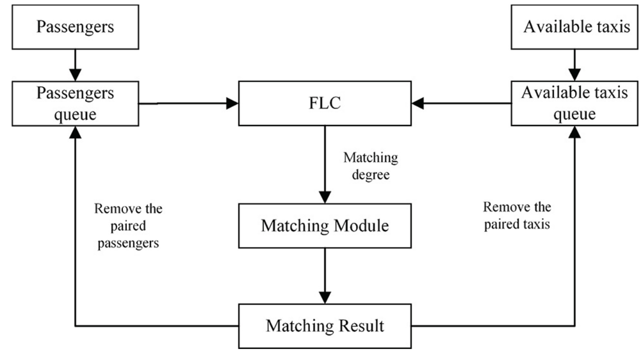


Fig. 2. Framework of the FLC-based taxi dispatch system.

$$\text{Expense} = \beta \cdot \sum_{t}^{T} \sum_{i \in \phi_t} \sum_{j \in \psi_t} H_t[i, j] \cdot (D_i + E(i, j)) \tag{14}$$

where D_i represents the required traverse distance of the passenger i; α is the revenue per kilometer; and β is the cost per kilometer. In this paper, we use $\alpha = 1.5$ and $\beta = 0.5$.

IV. FLC-Based Taxi Dispatch System

A. Framework of the FLC-Based Taxi Dispatch System

Fig. 2 illustrates the framework of the FLC-based taxi dispatch system. For each time frame, the unmatched passengers and the available taxis are input to the passenger queue and the taxi queue, respectively. Then, the relative attribute variables are input to the FLC. After that, the priority score (namely, the matching degree) for each possible taxi–passenger pair is calculated. Finally, the matching module performs the KM algorithm to pair the taxis and passengers according to the matching degrees. In the next of this section, we implement two subcomponents involved in the proposed system: 1) FLC and 2) KM.

B. FLC for Pairwise Prioritization

FLC mimics the decision making procedure of humans, which does not need an exactly analytic model. Typically, FLC consists of four parts: 1) a fuzzy rule base; 2) a fuzzification step; 3) an inference step; and 4) a defuzzification step [45]–[47]. Fig. 3 illustrates our proposed FLC in the pairwise prioritization module. In the FLC, the fuzzification step converts three crisp input attributes into the linguistic fuzzy states, each with a membership value calculated by the membership function. The inference step then maps the input fuzzy state to the output fuzzy state through the fuzzy rule base. Finally, the defuzzification step transforms each output fuzzy state into a crisp matching degree.

There also exists some more powerful fuzzy controllers like type-2 fuzzy. In [48], the type-2 fuzzy controller was used for the urban traffic network. The experiments validates the effectiveness and superiority of the type-2 fuzzy. In addition, type-2 fuzzy has also been applied into other difficult problems like classification, regression, etc. [49]–[51]. Besides, the analytic hierarchy process (AHP) mechanism [52], [53], which adapts the weighting factors of the input factors, is also a promising way to enhance the performance of the FLC. But in this paper,

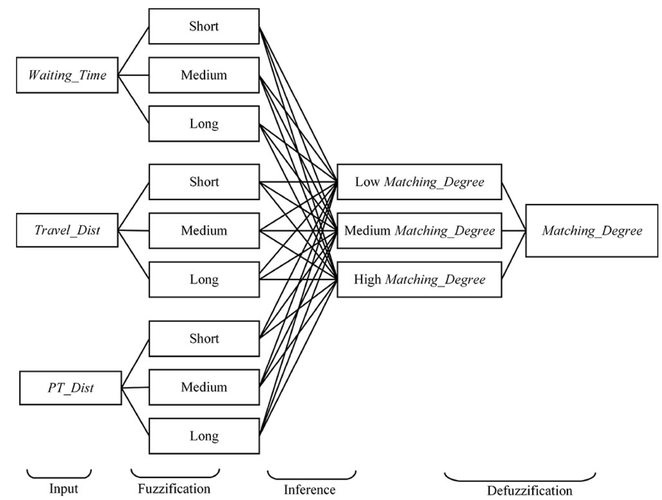


Fig. 3. Schematic of the FLC.

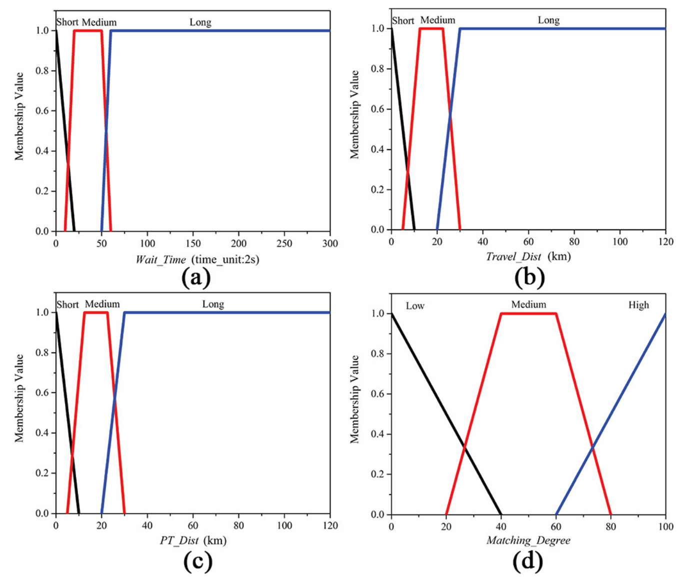


Fig. 4. Membership functions of the input and output attributes. (a) Membership functions of the *Waiting_Time.* (b) Membership functions of the *Travel_Dist.* (c) Membership functions of the *PT_Dist.* (d) Membership functions of the *Matching_Degree.*

we concentrate more on the practicality about using fuzzy controller to solve this RTTPM and consider the classical FLC. The details of the implementations of the FLC are stated as follows.

1) Input Variables: The FLC includes three input variables: 1) *Waiting_Time*; 2) *Travel_Dist*; and 3) *PT_Dist*. The *Waiting_Time* refers to the time duration since the passenger sent the booking request. The *Travel_Dist* denotes the origin to destination distance of the passenger. The *PT_Dist* denotes the distance between the passenger and taxi. The three input variables are chosen from entire process of a passenger hailing a taxi to get served. There are three steps to complete the service after the passenger sends the request. First, the system performs a matching procedure to designate a taxi to this passenger. Second, the taxi goes to the passengers location to pick up the passenger. Third, the taxi serves the passenger to the destination. The *Waiting_Time* of the passengers is mainly derived from the first and second steps. The *PT_Dist* is involved in the second step. In addition, the *Travel_Dist* is the most important factor in the final step. The *Waiting_Time* and *PT_Dist* are related to the QoS since the passengers want to get served in short time. The *Travel_Dist* and *PT_Dist* determine the profit of the taxi drivers. So, we consider this three attributes as the input variables in the proposed FLC, which are calculated as follows:

$$\begin{cases} Waiting\_Time = t - b_i \\ Travel\_Dist = D_i \\ PT\_Dist = E(i, j). \end{cases} \tag{15}$$

With the three variables incorporated into the FLC as above, the taxi–passenger matching degree takes multiple attributes that are most relevant to QoS and profit into consideration. Compared to the methods that only consider one attribute (e.g., NN or MP), the FLC calculates the matching degree in a more comprehensive manner, and thus helps to lower the risk of inducing extreme cases (e.g., passengers far from available taxi or having short traveling distance need to wait a long time or are even forced to cancel their requests).

2) Membership Functions: The membership functions of the three input and output variables are shown in Fig. 4. Three membership functions (*short*, *medium*, and *long*) are utilized for each input variables. In the fuzzification process, those membership functions are used to convert the crisp input values to the fuzzy linguistic values. The membership value is a unitless number ranging from 0 to 1, which describes the degree that the crisp input belongs to the fuzzy set. Besides, note that we use the trapezoidal membership functions here. Actually, some other shapes of membership functions, such as the Gaussian membership functions [54], [55], are also available to our proposed system. But the trapezoidal membership functions are the most efficient one owing to its simple calculation. Considering that the real-time taxi dispatch system requires taxi–passenger matching responses as quickly as possible, we adopt the trapezoidal membership functions in this paper.

3) Fuzzy Rule Base and Inference: The fuzzy rule base is usually determined by the expertise experience, which typically consists of a number of "IF-THEN" rules as

$$\begin{aligned} &\text{IF (a set of conditions is satisfied)} \\ &\text{THEN (a set of results can be inferenced).} \end{aligned}$$

Table II shows the fuzzy rule base of the FLC. There are 27 rules in the rule base. In this paper, two fuzzy set operators, intersection and union, are adopted. The intersection operator is the same as the min operator in mathematic. The union operator is the same as the max operator. In the rule matching process, for example, assumed that the membership value of "*Travel_Dist* is short" is m_1, the membership value of "*PT_Dist* is short" is m_2, and the membership value of "*Waiting_Time* is short" is m_3, then the strength of the first rule is calculated as $r1 = \min\ (m_1,\ m_2,\ m_3)$. On the other side, since the high matching state is inferenced from rules $r12$, $r20$, $r21$, and $r24$, the membership of "*Matching_Degree* is high" is calculated as $o_1 = \max\ (r12,\ r20,\ r21,\ r24)$.

TABLE II
FUZZY RULE BASE

Rule	*Travel_Dist*	*PT_Dist*	*Waiting_Time*	*Matching_Degree*
r1	short	short	short	low
r2	short	short	medium	medium
r3	short	short	long	medium
r4	short	medium	short	low
r5	short	medium	medium	low
r6	short	medium	long	medium
r7	short	long	short	low
r8	short	long	medium	low
r9	short	long	long	low
r10	medium	short	short	medium
r11	medium	short	medium	medium
r12	medium	short	long	high
r13	medium	medium	short	low
r14	medium	medium	medium	medium
r15	medium	medium	long	medium
r16	medium	long	short	low
r17	medium	long	medium	low
r18	medium	long	long	medium
r19	long	short	short	medium
r20	long	short	medium	high
r21	long	short	long	high
r22	long	medium	short	medium
r23	long	medium	medium	medium
r24	long	medium	long	high
r25	long	long	short	low
r26	long	long	medium	medium
r27	long	long	long	medium

4) Defuzzification: The defuzzification step transfers the fuzzy output values into a crisp output value. There are many methods that can be used in the defuzzification [56], [57]. A height defuzzification method [58] is used in this paper. In this method, the centroid of each membership function of the matching degree fuzzy set is calculated at first. Then, the final numerical output is calculated as follows:

$$O = \frac{\sum_{i=1}^{3} o_i \cdot a_i}{\sum_{i=1}^{3} o_i} \tag{16}$$

where o_i is the membership value of the ith fuzzy output and a_i is the abscissa of the centroid of the ith fuzzy output. Then, O is the final numerical output.

C. KM for Perfect Matching

After the FLC processing, a proximity matrix that contains the matching degrees of all possible taxi–passenger pair is input to the matching module. Unlike the traditional methods that adopt the FCFS strategy for matching, we use the KM algorithm [59], [60] to achieve the maximum weight perfect matching, which can improve the quality of the matching results. The matching degree matrix is regarded as a bipartite graph. Fig. 5 shows an example of the bipartite combinations of the passengers and taxis. If FCFS is applied, the total matching degree is $3+5+2 = 10$. In contrast, by the KM algorithm, the v_1 will be assigned to the p_3, the v_2 is matched to p_2, and the v_3 is going to serve p_1. The total matching degree increases to 13. So, the whole performance is improved.

V. PDE-BASED FUZZY LOGIC CONTROLLER

In the conventional FLC, the membership functions and fuzzy rule base are set by experts. The drawback of this method is that the expert cannot guarantee whether the membership functions and fuzzy rule base are optimal or not. For the RTTPM, there does not exist a sophisticated rule base for evaluating the matching degree. Without sufficient experience, one can hardly design an optimal rule base that helps to maximize the QoS and profit of the entire system (as required by our proposed work). So, the performance of the FLC still has room to improve. In the literature, many researches have applied the evolution algorithm to optimize the FLC. The experimental results proved the effectiveness of this method. However, most of them only optimized a single part of the two components (the membership functions, in most cases). If we can optimize the membership functions and the rule base together, the two critical factors in FLC are able to fit each other to obtain the optimal system performance. However, the optimization of the combination of the membership functions and fuzzy rule base is a mix-domain optimization problem. To overcome this challenge, we develop a new individual representation method and incorporate it into the PDE-based FLC. The method enables the simultaneous optimization of membership functions and the rule table. The details of the PDE-FLC are illustrated in the following sections.

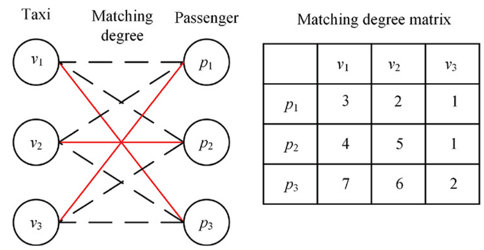


Fig. 5. Example of bipartite combinations of the passengers and taxis.

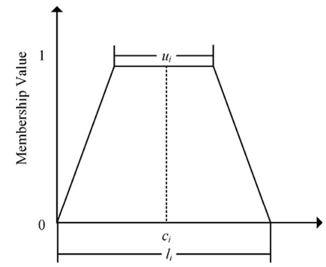


Fig. 6. Representation of trapezoidal functions.

A. Individual Representation and Initialization

To optimize the FLC, first, we need to define the individual structure and the search space. In the PDE-FLC, each of the three input attributes has three membership functions, while the output attribute also has three membership functions. Therefore, there are a total number of 12 membership functions to optimize. In this paper, each membership function is represented by a trapezoidal function. Fig. 6 shows the representation of the trapezoidal function: each membership function is represented as a triple array (u_i, l_i, c_i), where u_i is the length of the upper line, l_i denotes the length of the under line, and c_i is the middle point of the trapezoid. There exists a constraint of the triple array that the l_i must be larger than u_i.

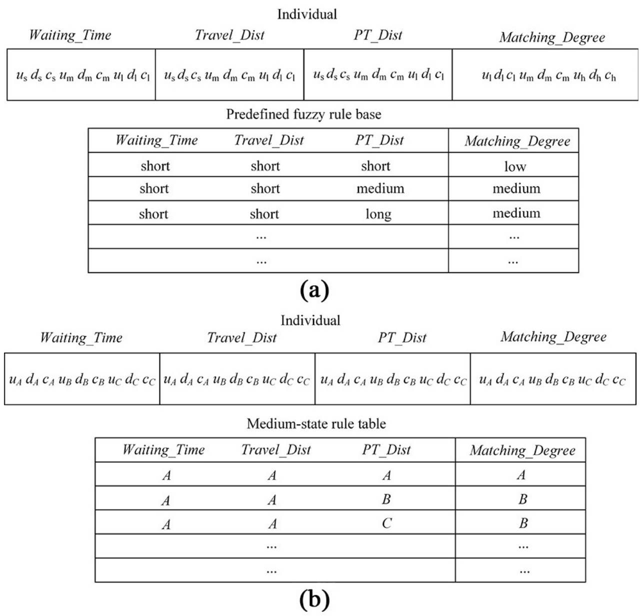


Fig. 7. Structure of the individual. (a) Conventional representation method. (b) New representation method.

To eliminate this constraint, we use a variable $d_i = l_i - u_i$ to replace the l_i in the triple array. This way, each membership function is represented as (u_i, d_i, c_i). Since we have 12 membership functions, the individual can be represented as $X_{i,g}$ = (u_1, d_1, c_1, u_2, d_2, $c_2, \ldots, u_{12}$, d_{12}, c_{12}), where g denotes the number of the current generation of the population. The ranges of all the parameters in the individuals are presented as follows:

$$\begin{aligned}
&0 < u_i < Max_Waiting_Time\\
&0 < d_i < 2 \cdot Max_Waiting_Time\\
&0 < c_i < Max_Waiting_Time, \qquad (i = 1, 2, 3)\\
&0 < u_i < Max_Travel_Dist\\
&0 < d_i < 2 \cdot Max_Travel_Dist\\
&0 < c_i < Max_Travel_Dist, \qquad (i = 4, 5, 6)\\
&0 < u_i < Max_PT_Dist\\
&0 < d_i < 2 \cdot Max_PT_Dist\\
&0 < c_i < Max_PT_Dist, \qquad (i = 7, 8, 9)\\
&0 < u_i < Max_Matching_Degree\\
&0 < d_i < 2 \cdot Max_Matching_Degree\\
&0 < c_i < Max_Matching_Degree, (i = 10, 11, 12).
\end{aligned}$$

By developing an association scheme, our method is able to optimize the membership functions and the fuzzy rule base together. The association scheme is described as follows. First, let us take a look at the representation used in the conventional EA-based FLC, which is illustrated in Fig. 7(a). Four segments in each individual stand for the four attributes. In each segment, there are three membership functions of the three fuzzy states (*short*, *medium*, and *long*), which can be parameterized as a triple array. There are some constraints to restrict an individual, since the order of the membership functions must be considered, e.g., the short function should be located left to the medium function as illustrated in Fig. 4. Then, all individuals in the population share the same fuzzy rule base, which is defined by some external effort. Differently, as shown in Fig. 7(b), our representation method does not restrain the order of the parameters. Instead, we use three state variables denoted as A, B, and C to represent any possible states of the fuzzy set. Meanwhile, each individual is companied with a medium-state rule table. After an individual is generated, we apply an association scheme to provide the meanings of the states A, B, and C based on the parameter distribution. For example, if the distribution of the three membership functions for the attribute *Waiting_Time* is $B < C < A$, then A, B, and C are associated with the long, short, and medium functions for the *Waiting_Time*, respectively. After determining the meanings of all membership functions for all attributes, the medium-state rule table is filled with the meaningful states, which becomes a conventional rule table for FLC. The benefit of this method is that different individuals may be associated with different fuzzy rule bases. Therefore, during the evolution process, the fuzzy rule base is optimized together with the membership functions. Using the new representation method, the initialization process is quite simple. All the individuals in the populations are generated randomly in the corresponding parameter ranges.

B. Fitness Evaluation

To evaluate the quality of individuals, a fitness function is necessary in DE. The fitness function is used to guide the evolution process, leading the population to converge to the optimal solution. For the RTTPM, as described in Section III, the QoS and the profit are the two objectives. Based on these, the fitness function is formulated as follows:

$$Fitness(SR, TW, PR) = w_1 \cdot SR + w_2 \cdot TW + w_3 \cdot PR \quad (17)$$

$$w_1 + w_2 + w_3 = 1 \quad (18)$$

where *SR*, *TW*, and *PR* are the three measures defined in Section III, which are obtained via the simulation of the taxi dispatch system. Since we need to maximize *SR* and *PR* and minimize *TW*, the *TW* term in (17) is prefixed with a negative sign. Different weights are tied to the three criteria to demonstrate the preference to each one. Higher weight means we care more about this criterion. To balance the influence of the three components, each component should be normalized before it is processed. A min–max normalization function is adopted to normalize those components in this paper

$$\text{Normalize}(x) = \frac{x - x_{\min}}{x_{\max} - x_{\min}}. \quad (19)$$

C. Parallel Differential Evolution

The process of the PDE is shown in Algorithm 1. First, the population is generated randomly. Then, the differential mutation and crossover operators are performed on the current generation. Afterward, a selection method is adopted to generate the next generation. In PDE, a parent vector from the current generation is called a target vector. In addition,

Algorithm 1 DE Algorithm

```
/* Initialization */
Generate the initial population of NP individuals and set g = 0;
/* Main Loop */
while Termination Rule is not met do
    Mutation by Eq.(20);
    Crossover by Eq.(21);
    Fitness Evaluation by Eq.(17);
    Selection by Eq.(22) to form the next generation;
    g++
end while
return The best individual.
```

a mutant vector obtained by the differential mutation operator is denoted as a donor vector. Then, the crossover operator recombines the target vector and the donor vector to generate an offspring named trial vector. Afterward, the trial vector competes with the target vector in the selection operator. In this paper, a simple form of the mutation operators is adopted. To form the donor vector for the ith target vector for each generation, three parameter vectors are randomly sampled from the current generation, which is denoted as $X_{r1,g}$, $X_{r2,g}$, and $X_{r3,g}$. The difference of $X_{r2,g}$ and $X_{r3,g}$ is scaled by the scalar parameter F, and the scaled differential vector is added to the third vector $X_{r1,g}$. Particularly, the donor vector $V_{i,g}$ is formulated as

$$V_{i,g} = X_{r1,g} + F \cdot (X_{r2,g} - X_{r3,g})(i \neq r1 \neq r2 \neq r3). \quad (20)$$

The crossover operator is performed after the mutation. The donor vector exchanges the genes with target vector to form a trial vector $U_{i,g}$. PDE adopts the binomial crossover operator, in which the trial vector inherits the genes from either the target vector or the donor vector according to a crossover rate Cr. This crossover method is formulated as

$$u_{ij,g} = \begin{cases} x_{ij,g}, & \text{if rand}(0,1) < \text{Cr or } j = j\text{rand} \\ v_{ij,g}, & \text{otherwise} \end{cases} \quad (21)$$

where $u_{ij,g}$ is the jth parameter in $U_{i,g}$, $x_{ij,g}$ is the jth parameter in the target vector $X_{i,g}$, and $v_{ij,g}$ is the jth parameter in the donor vector $V_{i,g}$. After the mutation and crossover, a selection operator based on the survival of fitness strategy [34] is performed to select the offspring. The fitness of the target vector is compared with that of the trial vector. The one with better fitness will be selected into the next generation. The selection is formulated as

$$X_{i,g+1} = \begin{cases} U_{i,g}, & \text{if } \mathit{Fitness}(U_{i,g}) > \mathit{Fitness}(X_{i,g}) \\ X_{i,g}, & \text{otherwise.} \end{cases} \quad (22)$$

Through the mutation, crossover, and selection operators, the individuals of PDE evolve according to the fitness function. Finally, after a number of generations, the population of individuals converges to the global optimum. So that the optimal membership functions and fuzzy rule base are arrived. In the above procedure, in order to evaluate the fitness of individuals, simulation is required to calculate the QoS and profit measures, namely, the *SR*, *TW*, and *PR* values. However, for large-scale problems, the evaluation cost could be very high, which leads to the long runtime of DE. To further improve the efficiency of the algorithm, we adopt a master–slave parallel model. The PDE-FLC is implemented by MPI, a message-passing system for parallelization of distributed computation [61], [62]. Fig. 8 demonstrates the flowchart of the PDE-FLC. The master node is responsible for performing the major DE operators, including initialization, mutation, crossover, and selection. The slave runs the simulation to obtain the fitness of the individuals. In this way, we speed up the execution of PDE.

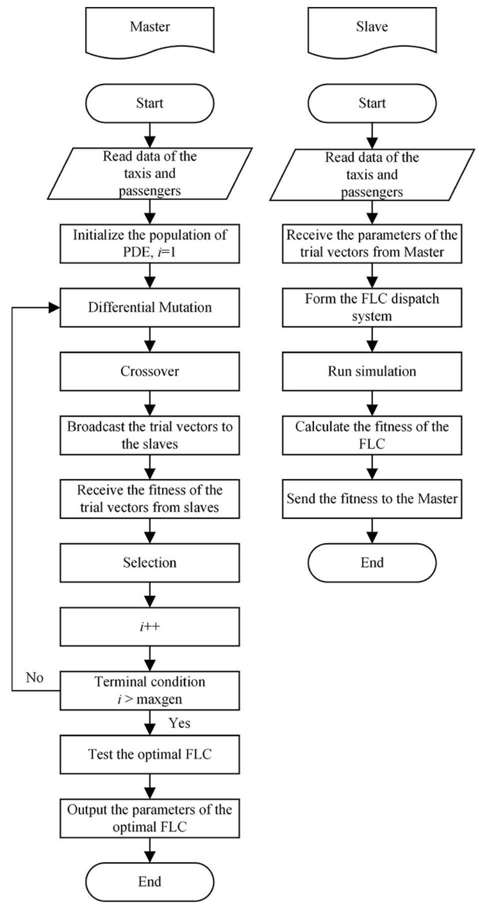


Fig. 8. Flowchart of the PDE-FLC.

VI. Simulation Result

This paper develops a taxi dispatch system, which is composed of a pairwise prioritization module using FLC and a matching module using KM. In addition, a PDE algorithm is developed to optimize the membership functions and rule base in FLC. The optimization is accomplished in an offline way, and then the system is performed in real time. In this section, first, investigation simulations are conducted to validate the effectiveness of different components of the proposed PDE-FLC system. Further, we compare our PDE-FLC with two recently proposed matching algorithms KMBA [7] and GPBM [10], which are originated from the methods of NN and MP, respectively. A normalized linear aggregation (NLA) algorithm is also compared in the simulations, which is commonly adopted to deal with multi-input situation [63]. Additionally, we exploit how the different weight settings in the fitness function influence the optimization results of FLC. The computational time of the PDE-FLC is also tested on each test case to validate the real-time characteristic of our method.

A. Simulation Setup

Simulations are performed based on the road network in the central area of Beijing. There are 20 000 locations that passengers can get on or off. We conduct two different kinds of test cases in our simulations, which are named as balance (B) and unbalance (UB), respectively. In the B test cases, the requests of the passengers are distributed in a uniformly random way. In the UB test cases, the time distribution of the passengers is quite unbalanced. Namely, the passenger demands vary over time, e.g., there are peak and off-peak time periods. Meanwhile, the spatial distribution of the taxis and passengers is also unbalanced in the UB test cases. This simulation strategy is quite similar to that in [64], where Grau *et al.* suggested to consider unbalanced distributions and the time-varying characteristics of passenger demands, since these situations are commonly seen in the real life. For all the test cases, 500 idle taxis are set in the road network at the beginning of the simulation. In addition, three different passenger scales, namely, 3000, 4000, and 5000 passengers, are used to test the performance of the matching algorithms under different working pressure. The corresponding test cases in B and UB are denoted as B1, B2, and B3 and UB1, UB2, and UB3, respectively.

PDE adopts the parameter settings that are suggested in [34] and [35]: the maximum generation is 200, the population size is 100, the crossover rate *Cr* is 0.5, and the scalar factor F is 0.5. w_1, w_2, and w_3 are 0.4, 0.1, and 0.5, respectively, for the fitness evaluation. The PDE-FLC is first offline trained on a Linux cluster which has 68 cores at 1.4 GHz with 112-GB RAM. Note that 51 cores are used in our PDE, one for the master and the other 50 cores for running the slave procedures. After the offline training, the simulations are conducted on single core with about 4-GB RAM requirement which is easily satisfied in the real world. In [10], the sampling time between two matching procedure has been suggested to be 2–20 s. In the simulations, we set the sampling time to 2 s. Based on this setting, the total number of the time frames is 1800 for each hour. All the test cases simulate one hour of real-time passengers-taxis matching. The matching methods are conducted in every time unit to provide real-time responses (solutions) to the RTTPM.

In the simulations, we use the service rate *SR*, time of waiting *TW*, profit *PR*, and *Fitness* to measure the performance of the algorithms. The *SR* measures how much pressure the taxi dispatch system can bear. The *TW* measures the average waiting time of passengers. This two measurements are related to the QoS. As for the *PR*, it indicates the profit that the company can gain. The *Fitness* measures the comprehensive performance of the system based on the above three measurements.

TABLE III
COMPARISONS BETWEEN FLC-FCFS AND FLC-KM

Test case	Measurement	FLC-FCFS	FLC-KM
B1	*SR*	**0.501**	0.453
	TW	209.3	**182.7**
	PR	10189.3	**12605.1**
	Fitness	0.3552	**0.4133**
B2	*SR*	**0.383**	0.343
	TW	317.9	**257.5**
	PR	10778.9	**13780**
	Fitness	0.3169	**0.3958**
B3	*SR*	**0.311**	0.279
	TW	327.1	**266.8**
	PR	10770	**14088.1**
	Fitness	0.2847	**0.3748**
UB1	*SR*	**0.514**	0.467
	TW	293.8	**258.2**
	PR	10742.5	**12608.9**
	Fitness	0.3762	**0.4159**
UB2	*SR*	**0.401**	0.36
	TW	307.1	**269.5**
	PR	10779	**13428.4**
	Fitness	0.3274	**0.3897**
UB3	*SR*	**0.315**	0.282
	TW	318.1	**273.1**
	PR	10990.6	**13960.5**
	Fitness	0.2949	**0.3707**

B. Validation of the KM Component in FLC

In many traditional taxi dispatch system, the FCFS strategy is adopted to handle the requests of passengers. Namely, the passengers who sent requests earlier will be assigned with a higher priority to choose the taxis. Differently, in our proposed system, the KM algorithm is adopted in the matching module to enhance the performance of the matching process. Now, we conduct simulations to compare these two matching strategies, while the other components are kept as the same in the FLC taxi dispatch system. Table III shows the results. It can be observed that KM outperforms FCFS in all the test cases. For the FLC-FCFS, the passengers that come first will be served first, so that the service rate *SR* is better than the FLC-KM. However, the FCFS does not consider the global matching effect that it may lose some high-quality deals. On the contrary, using the global KM matching algorithm, the FLC-KM can maximize the overall matching degree in each matching procedure. So, the comprehensive performance of FLC-KM is better than the FLC-FCFS. The *Fitness* of KM is better than the FCFS, which also reveals the advantage of the KM. Therefore, adopting the KM algorithm in the matching module helps to improve the system performance.

C. Validation of the PDE Component in PDE-FLC

The membership functions and the fuzzy rule base are two crucial parts in the FLC. In the FLC-based taxi dispatch system, if the membership functions are not optimal or the fuzzy rule base is not reasonable, the performance of the system could be unsatisfactory. Using the expert experience to determine the membership functions and fuzzy rule base is tedious work, and, what is worse, it is not always reliable. In this section, we compare FLC with the PDE-FLC to investigate the effectiveness of the PDE component. Besides, we also include a CPDE-FLC that optimizes the membership

TABLE IV
COMPARISONS AMONG FLC, CPDE-FLC, AND PDE-FLC

Test case	Measurement	FLC	CPDE-FLC	PDE-FLC
B1	*SR*	0.453	0.581	**0.661**
	TW	249.6	135.9	**112.2**
	PR	12605.1	14996.9	**15298.1**
	Fitness	0.4133	0.5621	**0.6102**
B2	*SR*	0.343	0.459	**0.536**
	TW	257.6	139.9	**119.7**
	PR	13780	**16624.8**	16311.6
	Fitness	0.3958	0.5527	**0.5821**
B3	*SR*	0.279	0.38	**0.46**
	TW	266.8	140.7	**117.8**
	PR	14088.1	**17090.3**	16544.2
	Fitness	0.3748	0.5325	**0.5682**
UB1	*SR*	0.467	0.599	**0.666**
	TW	258.2	147.5	**123.4**
	PR	12608.9	15370.4	**15611**
	Fitness	0.4159	0.5749	**0.6155**
UB2	*SR*	0.36	0.492	**0.565**
	TW	269.5	145.2	**128.6**
	PR	13428.4	**16596**	16490.2
	Fitness	0.3897	0.5632	**0.6012**
UB3	*SR*	0.2818	0.3832	**0.466**
	TW	273.1	145.1	**129.1**
	PR	13960.5	**17263.2**	16982.8
	Fitness	0.3707	0.5365	**0.5779**

TABLE V
RESULTS OF DIFFERENT ALGORITHMS FOR RTTPM

Test case	Measurement	GPBM	KMBA	NLA	PDE-FLC
B1	*SR*	0.492	0.625	0.505	**0.661**
	TW	139.2	112.2	177.8	**112.1**
	PR	14766.7	**15593**	14979.1	15298.1
	Fitness	0.5197	0.6024	0.5172	**0.6102**
B2	*SR*	0.356	0.487	0.362	**0.5355**
	TW	148.5	121.2	198.6	**119.8**
	PR	15932	**16831.2**	15938.1	16311.6
	Fitness	0.4913	0.5752	0.477	**0.5821**
B3	*SR*	0.281	0.407	0.288	**0.46**
	TW	146.7	118.4	208.4	**117.8**
	PR	16397.9	**17427.6**	16601.5	16544.2
	Fitness	0.4735	0.5590	0.4609	**0.5682**
UB1	*SR*	0.506	0.634	0.513	**0.666**
	TW	167.8	**123.1**	197.2	123.4
	PR	14891.5	**15794.2**	15141.9	15611
	Fitness	0.5189	0.5913	0.5179	**0.6155**
UB2	*SR*	0.369	0.521	0.377	**0.565**
	TW	159.3	**125.1**	214.5	128.7
	PR	15872.5	**16993.3**	16235.5	16490.2
	Fitness	0.4912	0.5913	0.4851	**0.6012**
UB3	*SR*	0.284	0.415	0.288	**0.466**
	TW	155.3	129.9	217.6	**129.1**
	PR	16684.2	**17576.3**	16996.4	16982.8
	Fitness	0.4790	0.5620	0.4675	**0.5779**

functions only in the comparison. Since PDE-FLC optimizes the membership functions and the rule base together, the comparison between PDE-FLC and CPDE-FLC helps to reveal the effectiveness of our new optimization method. The matching module applies the KM algorithm to pair the taxis and passengers. To make the results convincing, the optimization component of PDE-FLC and CPDE-FLC is conducted on an extra training case instead of the test cases. Then, the test cases are used to evaluate the performance of different algorithms.

The results are shown in Table IV. It can be observed that PDE-FLC outperforms the FLC significantly. The *SR*, *TW*, and *PR* have been improved a lot. It validates the effectiveness of applying the PDE to optimize the FLC module in the taxi dispatch system. In addition, the PDE-FLC also outperforms the CPDE-FLC that only optimizes the membership functions of the FLC. It can be observed that, for all the instances, PDE-FLC enhances both QoS and profit. It proves that optimizing the membership functions and fuzzy rule base of the FLC simultaneously is effective and can greatly improve the performance.

D. Comparison With the Other Algorithms

Further, we compare the PDE-FLC with two recently proposed greedy algorithms for taxi–passenger matching, namely KMBA [7] and GPBM [10]. The KMBA combines the greedy strategy of NN with the KM algorithm, whilst the GPBM combines the greedy strategy of MP with the packing-based matching mechanism. The above two greedy algorithms are selected as they represent the two most commonly used greedy strategies, NN and MP, respectively. In addition, we involve another greedy algorithm named NLA into comparison. In the NLA, the matching degree is computed as the sum of the normalized values of the three input variables in the fuzzy rule base of PDE-FLC, i.e., *Travel_Dist*, *PT_Dist*, and *Waiting_Time*. Based on the matching degrees, the KM algorithm is applied to achieve the maximum weight perfect matching. Comparing the results of PDE-FLC with the above three greedy algorithms, we can not only validate the effectiveness of the PDE-FLC but also can reveal whether simultaneous consideration of multiple attributes in matching and the usage of FLC can enhance the algorithmic performance.

Table V presents the results of KMBA, GPBM, NLA, and PDE-FLC for RTTPM. It can be observed that although the greedy algorithms can produce valid solutions, PDE-FLC generally outperforms them in terms of *SR*, *TW*, and *PR*. For GBPM, the use of the max-profit greedy strategy makes it favor passengers with longer travel distance. As the requests of passengers with short travel distance cannot be satisfied, the service rate of GBPM is often the lowest among all the four algorithms. In NLA, the linear aggregation of the input attributes may cause their values to cancel out each other and thus result in inaccurate matching degrees, which leads to its poor performance. KMBA performs the best among the three greedy algorithms. The NN greedy strategy allows it to prioritize matching between nearer taxi and passengers, which helps to reduce the passengers waiting time and increase the service rate. Further, the increase in service rate improves the total profit. However, the proposed PDE-FLC remains advantageous over KMBA in terms of *SR* and *TW*, especially in the UB test cases. The fitness values of KMBA and PDE-FLC also indicate that PDE-FLC achieves a better overall performance. To summarize, PDE-FLC is effective in dealing with the RTTPM and can provide promising profit and QoS values simultaneously.

E. Exploration of Weight Settings

In the PDE-FLC, the fitness function guides the optimization in the evolution process. In the fitness function,

TABLE VI
RESULTS OF PDE-FLC UNDER DIFFERENT WEIGHT SETTINGS

Test case	Measurement	WS_1	WS_2	WS_3
B1	*SR*	0.615	0.6606	**0.691**
	TW	**102.5**	112.2	109.1
	PR	**15593.3**	15298.1	15205.2
B2	*SR*	0.478	0.536	**0.548**
	TW	**107.1**	119.8	114.3
	PR	**16802.4**	16311.6	16089.1
B3	*SR*	0.401	0.4598	**0.481**
	TW	**103.6**	1178	113.5
	PR	**17578.8**	16544.2	16502.1
UB1	*SR*	0.619	0.666	**0.701**
	TW	**113.8**	123.4	121.4
	PR	15581.3	**15611**	15503.5
UB2	*SR*	0.507	0.565	**0.580**
	TW	**110.8**	128.7	125.2
	PR	**16802.5**	16490.2	16376.8
UB3	*SR*	0.404	0.466	**0.488**
	TW	**112.7**	129.1	126.8
	PR	**17542.5**	16982.8	17008

we set different weight to each criterion, which shows how much preference to that criterion. Three different weight settings are adopted to explore the influence in optimizing the FLC. WS_1, WS_2, and WS_3 represent the preference of profit, no preference, and preference of QoS, respectively,

$$\text{WS}_1\{w_1 : 0.2, w_2 : 0.1, w_3 : 0.7\}$$
$$\text{WS}_2\{w_1 : 0.4, w_2 : 0.1, w_3 : 0.5\}$$
$$\text{WS}_3\{w_1 : 0.6, w_2 : 0.1, w_3 : 0.3\}.$$

In Table VI, for the fewer passenger cases, B1 and UB1, the changes of the weights do not obviously influence the performance, because the taxis are almost enough to serve those passengers. When the passenger requests increase, the service rate increases when WS_3 is applied to the fitness function (which assigns the highest weight to the *SR*). When WS_2 is adopted, the optimized FLC shows a good balance in the QoS and profit. The weight setting WS_1 guides the PDE to optimize an FLC that cares more about the profit. The simulation results also validate that the profit of that case is the highest. In the conclusion, for the fewer passenger situations, different weight settings do not influence the FLC too much. But, as the number of the passenger increases, the influence becomes more prominent. So, the taxi service provider can choose different weight settings based on their own preferences to make a flexible control of FLC to cope with different situations, which is another advantage of the PDE-FLC.

F. Computational Time

The total response time of the proposed system is mainly influenced by the data collection time, the computational time of the proposed PDE-FLC, and the diffusion time of the scheduling results. Among these three terms, the first and third terms are largely dependent of the platform that the system runs on and thus is not analyzed here. We focus on analyzing the computational time of PDE-FLC. To do so, we record the average execution time of each matching procedure. As presented in Fig. 9, the PDE-FLC takes only 4.6–20.1 ms to

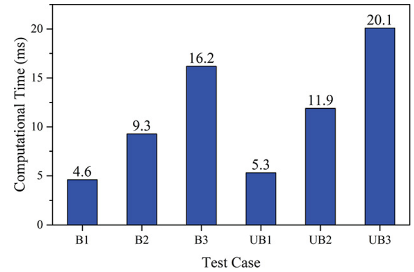


Fig. 9. Computational time of PDE-FLC for different cases.

calculate the matching result under different working pressure. The time is far less than the sampling time (2 s), which cannot even be noticed by humans. The results prove that the PDE-FLC can respond quickly and satisfy the real time requirement in the real-world situations.

VII. CONCLUSION

In this paper, we design a two-stage taxi–passenger matching system to solve the RTTPM. The system applies the FLC for pairwise prioritization, and it then adopts the KM algorithm for pairwise matching. However, the traditional FLC is not always effective and reliable, since the membership functions and fuzzy rule base are not optimal. To tackle this problem, we propose the PDE to optimize the FLC. The PDE-FLC embeds a new individual structure with an association scheme, which enables the concurrent optimization of the membership functions and the fuzzy rule base. To reduce the execution time of the evolution process, the proposed algorithm is implemented in a parallel way using the master–slave model.

Several simulations were conducted to explore the performance of the PDE-FLC. The simulation results validated the effectiveness of the proposed KM, FLC, and the PDE optimization components in improving the quality of RTTPM. We also showed that it is useful for performance enhancement by optimizing the membership functions and fuzzy rule base together. The entire PDE-FLC system showed advantages to enhance the QoS while maintaining good profits, when compared with the previous algorithms. In addition, the PDE-FLC can flexibly deal with different passenger situations through adjusting the weight settings in the fitness functions of PDE. To summarize, we developed the PDE-FLC, which is an effective and flexible schedule algorithm for the modern taxi–passenger matching system.

For the future work, we will explore some more powerful fuzzy controllers such as the type-2 fuzzy controller, which may enhance the performance of the basic FLC adopted in this paper. The AHP mechanism can also be explored for further performance boost. In addition, because the RTTPM has multiobjective optimization feature, some evolutionary multiobjective algorithms (MOEA/D, HypeE, and NSGA-III) will also be considered to extend this paper.

REFERENCES

[1] H. Yang, M. Ye, W. H. Tang, and S. C. Wong, "Regulating taxi services in the presence of congestion externality," *Transport. Res. A Policy Pract.*, vol. 39, no. 1, pp. 17–40, 2005.

[2] M. Nourinejad and M. Ramezani, "Developing a large-scale taxi dispatching system for urban networks," in *Proc. IEEE 19th Int. Conf. Intell. Transport. Syst. (ITSC)*, Nov. 2016, pp. 441–446.

[3] W. Zhang, R. He, Q. Xiao, and C. Ma, "Taxi carpooling model and carpooling effects simulation," *Int. J. Simulat. Process Modell.*, vol. 12, no. 3–4, p. 338, 2017.

[4] Z. Liao, "Real-time taxi dispatching using global positioning systems," *Commun. ACM*, vol. 46, no. 5, pp. 81–83, 2003.

[5] Y. Jianxin, Z. Xiaomin, and Z. Hongyu, "Design and implementation of taxi calling and dispatching system based on GPS mobile phone," in *Proc. 4th Int. Conf. Comput. Sci. Educ.*, Jul. 2009, pp. 1163–1169.

[6] C. Tian, Y. Huang, Z. Liu, F. Bastani, and R. Jin, "Noah: A dynamic ridesharing system," in *Proc. ACM SIGMOD Int. Conf. Manag. Data*, 2013, pp. 985–988.

[7] G. Gao, M. Xiao, and Z. Zhao, "Optimal multi-taxi dispatch for mobile taxi-hailing systems," in *Proc. 45th Int. Conf. Parallel Process. (ICPP)*, Aug. 2016, pp. 294–303.

[8] M. Meghjani and K. Marczuk, "A hybrid approach to matching taxis and customers," in *Proc. IEEE Region 10 Conf. (TENCON)*, Nov. 2016, pp. 167–169.

[9] M. Maciejewski, J. Bischoff, and K. Nagel, "An assignment-based approach to efficient real-time city-scale taxi dispatching," *IEEE Intell. Syst.*, vol. 31, no. 1, pp. 68–77, Jan./Feb. 2016.

[10] L. Zheng, L. Chen, and J. Ye, "Order dispatch in price-aware ridesharing," in *Proc. Very Large Database Endowment (PVLDB)*, vol. 11, no. 8, pp. 853–865, Apr. 2018.

[11] F. Miao *et al.*, "Taxi dispatch with real-time sensing data in metropolitan areas: A receding horizon control approach," *IEEE Trans. Autom. Sci. Eng.*, vol. 13, no. 2, pp. 463–478, Apr. 2016.

[12] K. T. Seow, N. H. Dang, and D.-H. Lee, "A collaborative multiagent taxi-dispatch system," *IEEE Trans. Autom. Sci. Eng.*, vol. 7, no. 3, pp. 607–616, Jul. 2010.

[13] M. Shrivastava, P. K. Chande, A. S. Monga, and H. Kashiwagi, "Taxi despatch: A fuzzy rule approach," in *Proc. Conf. Intell. Transport. Syst.*, Nov. 1997, pp. 978–982.

[14] M. N. Ngo, K. T. Seow, and K. W. Wong, "Fuzzy linear assignment problem: An approach to vehicle fleet deployment," in *Proc. IEEE Int. Conf. Fuzzy Syst.*, vol. 2, Jul. 2004, pp. 1197–1202.

[15] I. Kouatli, "Implementation of a multivariable modular structure for fuzzy taxi scheduling system (FTSS)," in *Proc. Int. Conf. Connect. Veh. Expo (ICCVE)*, Nov. 2014, pp. 959–965.

[16] Y. Ge, H. Xiong, A. Tuzhilin, K. Xiao, M. Gruteser, and M. Pazzani, "An energy-efficient mobile recommender system," in *Proc. ACM SIGKDD Int. Conf. Knowl. Disc. Data Min.*, Washington, DC, USA, Jul. 2010, pp. 899–908.

[17] S. Qian, Y. Zhu, and M. Li, "Smart recommendation by mining large-scale GPS traces," in *Proc. Wireless Commun. Netw. Conf.*, 2012, pp. 3267–3272.

[18] S. Qian, J. Cao, F. Le Mouël, I. Sahel, and M. Li, "SCRAM: A sharing considered route assignment mechanism for fair taxi route recommendations," in *Proc. ACM SIGKDD Int. Conf. Knowl. Disc. Data Min.*, 2015, pp. 955–964.

[19] H. Ying, Y. Ding, S. Li, and S. Shao, "Comparison of necessary conditions for typical Takagi–Sugeno and Mamdani fuzzy systems as universal approximators," *IEEE Trans. Syst., Man, Cybern. A, Syst., Humans*, vol. 29, no. 5, pp. 508–514, Sep. 1999.

[20] F. A. Márquez, A. PeregrÍn, and F. Herrera, "Cooperative evolutionary learning of linguistic fuzzy rules and parametric aggregation connectors for Mamdani fuzzy systems," *IEEE Trans. Fuzzy Syst.*, vol. 15, no. 6, pp. 1162–1178, Dec. 2007.

[21] A. Aceves-Lopez and J. Aguilar-Martin, "A simplified version of Mamdani's fuzzy controller: The natural logic controller," *IEEE Trans. Fuzzy Syst.*, vol. 14, no. 1, pp. 16–30, Feb. 2006.

[22] R. Sakthivel, P. Selvaraj, and B. Kaviarasan, "Modified repetitive control design for nonlinear systems with time delay based on T–S fuzzy model," *IEEE Trans. Syst., Man, Cybern., Syst.*, to be published.

[23] X. Xie, Q. Zhou, D. Yue, and H. Li, "Relaxed control design of discrete-time Takagi–Sugeno fuzzy systems: An event-triggered real-time scheduling approach," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 48, no. 12, pp. 2251–2262, Dec. 2018.

[24] J.-W. Kim and B. M. Kim, "A GA-based fuzzy traffic simulation for crossroad management," in *Proc. Congr. Evol. Comput.*, vol. 2, 2001, pp. 1289–1295.

[25] W. Wei and M.-J. Wang, "Fuzzy-GA-based traffic signal control," in *Proc. Int. Conf. Mach. Learn. Cybern.*, vol. 1, Nov. 2003, pp. 645–650.

[26] M. Collotta, G. Pau, and V. Maniscalco, "A fuzzy logic approach by using particle swarm optimization for effective energy management in IWSNs," *IEEE Trans. Ind. Electron.*, vol. 64, no. 12, pp. 9496–9506, Dec. 2017.

[27] S.-H. Tsai and Y.-W. Chen, "A novel fuzzy identification method based on ant colony optimization algorithm," *IEEE Access*, vol. 4, pp. 3747–3756, 2016.

[28] M. Antonelli, D. Bernardo, H. Hagras, and F. Marcelloni, "Multiobjective evolutionary optimization of type-2 fuzzy rule-based systems for financial data classification," *IEEE Trans. Fuzzy Syst.*, vol. 25, no. 2, pp. 249–264, Apr. 2017.

[29] Z. Kai, Y. J. Gong, and J. Zhang, "Real-time traffic signal control with dynamic evolutionary computation," in *Proc. IIAI 3rd Int. Conf. Adv. Appl. Informat.*, Aug. 2014, pp. 493–498.

[30] A. Ullah, J. Li, A. Hussain, and Y. Shen, "Genetic optimization of fuzzy membership functions for cloud resource provisioning," in *Proc. IEEE Symp. Comput. Intell. (SSCI)*, Dec. 2016, pp. 1–8.

[31] J. Xiu, C. Xia, and H. Fang, "GA-based adaptive fuzzy logic controller for switched reluctance motor drive," in *Proc. 6th World Congr. Intell. Control Autom.*, vol. 2, 2006, pp. 8226–8230.

[32] B. Suntaranont, C. Aryupong, S. Jankoo, and P. Champrasert, "Energy aware flash flood monitoring stations using a GA-fuzzy logic control mechanism," in *Proc. IEEE 10th Int. Conf. Intell. Sensors Sensor Netw. Inf. Process. (ISSNIP)*, Apr. 2015, pp. 1–6.

[33] A. Nabi and N. A. Singh, "GA optimization of fuzzy logic controller for voltage sag improvement in power systems," in *Proc. 10th Int. Conf. Intell. Syst. Control (ISCO)*, Jan. 2016, pp. 1–5.

[34] S. Das and P. N. Suganthan, "Differential evolution: A survey of the state-of-the-art," *IEEE Trans. Evol. Comput.*, vol. 15, no. 1, pp. 4–31, Feb. 2011.

[35] J. Wang, G. Liang, and J. Zhang, "Cooperative differential evolution framework for constrained multiobjective optimization," *IEEE Trans. Cybern.*, vol. 49, no. 6, pp. 2060–2072, Jun. 2019.

[36] A. K. Qin, V. L. Huang, and P. N. Suganthan, "Differential evolution algorithm with strategy adaptation for global numerical optimization," *IEEE Trans. Evol. Comput.*, vol. 13, no. 2, pp. 398–417, Apr. 2009.

[37] S. Das, A. Abraham, U. K. Chakraborty, and A. Konar, "Differential evolution using a neighborhood-based mutation operator," *IEEE Trans. Evol. Comput.*, vol. 13, no. 3, pp. 526–553, Jun. 2009.

[38] F. Xue, A. C. Sanderson, and R. J. Graves, "Multiobjective evolutionary decision support for design–supplier–manufacturing planning," *IEEE Trans. Syst., Man, Cybern. A, Syst., Humans*, vol. 39, no. 2, pp. 309–320, Mar. 2009.

[39] S. Das, A. Abraham, and A. Konar, "Automatic clustering using an improved differential evolution algorithm," *IEEE Trans. Syst., Man, Cybern. A, Syst., Humans*, vol. 38, no. 1, pp. 218–237, Jan. 2008.

[40] S.-Y. Chen and M.-H. Song, "Energy-saving dynamic bias current control of active magnetic bearing positioning system using adaptive differential evolution," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 49, no. 5, pp. 942–953, May 2019.

[41] Y. Fu, M. Ding, C. Zhou, and H. Hu, "Route planning for unmanned aerial vehicle (UAV) on the sea using hybrid differential evolution and quantum-behaved particle swarm optimization," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 43, no. 6, pp. 1451–1465, Nov. 2013.

[42] S. K. Verma and H. T. Vo, "A predictive taxi dispatching system for improved user satisfaction and taxi utilization," in *Proc. IEEE Int. Conf. Smart City SocialCom SustainCom (SmartCity)*, Dec. 2015, pp. 175–182.

[43] L. Moreira-Matias, J. Gama, M. Ferreira, J. Mendes-Moreira, and L. Damas, "Predicting taxi–passenger demand using streaming data," *IEEE Trans. Intell. Transp. Syst.*, vol. 14, no. 3, pp. 1393–1402, Sep. 2013.

[44] K. Zhang, K. Zhang, S. Leng, and S. Xu, "Adaptive airport taxi dispatch algorithm based on PCA-WNN," in *Proc. IEEE 11th Int. Conf. Depend. Auton. Secure Comput.*, Dec. 2013, pp. 340–343.

[45] J. Qiao, N. Yang, and J. Gao, "Two-stage fuzzy logic controller for signalized intersection," *IEEE Trans. Syst., Man, Cybern. A, Syst., Humans*, vol. 41, no. 1, pp. 178–184, Jan. 2011.

[46] S. Pandey, P. Mathur, and T. Patil, "Real time traffic signal control using fuzzy logic controller: Review," in *Proc. 1st Int. Conf. Intell. Syst. Inf. Manag. (ICISIM)*, Oct. 2017, pp. 307–313.

[47] C. C. Lee, "Fuzzy logic in control systems: Fuzzy logic controller. I," *IEEE Trans. Syst., Man, Cybern.*, vol. 20, no. 2, pp. 404–418, Mar./Apr. 1990.

[48] M. H. Khooban, N. Vafamand, A. Liaghat, and T. Dragicevic, "An optimal general type-2 fuzzy controller for urban traffic network," *ISA Trans.*, vol. 66, pp. 335–343, Jan. 2017.

[49] E.-H. Kim, S.-K. Oh, and W. Pedrycz, "Design of reinforced interval type-2 fuzzy c-means-based fuzzy classifier," *IEEE Trans. Fuzzy Syst.*, vol. 26, no. 5, pp. 3054–3068, Oct. 2018.

[50] I. Eyoh, R. John, G. D. Maere, and E. Kayacan, "Hybrid learning for interval type-2 intuitionistic fuzzy logic systems as applied to identification and prediction problems," *IEEE Trans. Fuzzy Syst.*, vol. 26, no. 5, pp. 2672–2685, Oct. 2018.

[51] Q.-F. Liao and D. Sun, "Interaction measures for control configuration selection based on interval type-2 Takagi–Sugeno fuzzy model," *IEEE Trans. Fuzzy Syst.*, vol. 26, no. 5, pp. 2510–2523, Oct. 2018.

[52] I. Kouatli, "A biologicaly inspired decision model for multivariable genetic- fuzzy-AHP system," *Procedia Comput. Sci.*, vol. 22, no. 22, pp. 2–9, 2013.

[53] I. Kouatli, "Fuzzimetric employee evaluations system (FEES): A multivariable-modular approach," *J. Intell. Fuzzy Syst.*, vol. 35, no. 4, pp. 4717–4729, Sep. 2018.

[54] V. Kreinovich, C. Quintana, and L. Reznik, "Gaussian membership functions are most adequate in representing uncertainty in measurements," in *Proc. North Amer. Fuzzy Inf. Process. Soc. Conf. (NAFIPS)*, Puerto Vallarta, Mexico, 1992, pp. 618–624.

[55] J.-S. R. Jang, "ANFIS: Adaptive-network-based fuzzy inference system," *IEEE Trans. Syst., Man, Cybern.*, vol. 23, no. 3, pp. 665–685, May/Jun. 1993.

[56] J. J. Saade and H. B. Diab, "Defuzzification techniques for fuzzy controllers," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 30, no. 1, pp. 223–229, Feb. 2000.

[57] W. Kosiński, P. Prokopowicz, and A. Rosa, "Defuzzification functionals of ordered fuzzy numbers," *IEEE Trans. Fuzzy Syst.*, vol. 21, no. 6, pp. 1163–1169, Dec. 2013.

[58] G. C. D. Sousa and B. K. Bose, "A fuzzy set theory based control of a phase-controlled converter DC machine drive," in *Proc. Conf. Rec. IEEE Ind. Appl. Soc. Annu. Meeting*, vol. 1, Sep. 1991, pp. 854–861.

[59] H. W. Kuhn, "The Hungarian method for the assignment problem," *Naval Res. Logist.*, vol. 2, nos. 1–2, pp. 83–97, 1955.

[60] J. Munkres, "Algorithms for the assignment and transportation problems," *J. Soc. Ind. Appl. Math.*, vol. 5, no. 1, pp. 32–38, 1957.

[61] Y.-J. Gong *et al.*, "Distributed evolutionary algorithms and their models: A survey of the state-of-the-art," *Appl. Soft Comput.*, vol. 34, pp. 286–300, Sep. 2015.

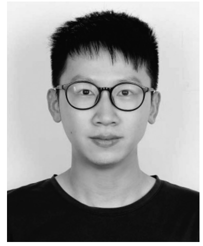
[62] Z.-H. Zhan *et al.*, "Cloudde: A heterogeneous differential evolution algorithm and its distributed cloud version," *IEEE Trans. Parallel Distrib. Syst.*, vol. 28, no. 3, pp. 704–716, Mar. 2017.

[63] Y.-W. Liu, X.-Y. Zhang, Y.-J. Gong, W.-N. Chen, and J. Zhang, "A parallel genetic algorithm with region division strategy to solve taxi-passenger matching problem," in *Proc. IEEE Symp. Comput. Intell. (SSCI)*, Nov./Dec. 2017, pp. 1–7.

[64] J. M. S. Grau *et al.*, "Informed versus non-informed taxi drivers: Agent-based simulation framework for assessing their performance," in *Proc. Transport. Res. Board 97th Annu. Meeting*, Washington, DC, USA, Jan. 2018, paper 18-04235.

Yue-Jiao Gong (S'10–M'15) received the B.S. and Ph.D. degrees in computer science from Sun Yat-sen University, Guangzhou, China, in 2010 and 2014, respectively.

She is currently a Professor with the School of Computer Science and Engineering, South China University of Technology, Guangzhou. Her current research interests include evolutionary computation, swarm intelligence, and their applications to intelligent transportation and smart city scheduling. She has published over 70 papers, including over 30 IEEE Transactions papers, in the above areas.



Yi-Wen Liu received the bachelor's degree in electronic information science and technology from the University of Electronic Science and Technology of China, Chengdu, China, in 2016. He is currently pursuing the master's degree with the South China University of Technology, Guangzhou, China.

His current research interests include computation intelligence, fuzzy system, and their real-world applications.

Ying Lin (M'13) received the Ph.D. degree in computer science from Sun Yat-sen University, Guangzhou, China, in 2012.

She is currently an Associate Professor with the Department of Psychology, Sun Yat-sen University. Her current research interests include computational intelligence and its applications in psychometrics and neuroimaging.

Wei-Neng Chen (S'07–M'12–SM'17) received the bachelor's and Ph.D. degrees in computer science from Sun Yat-sen University, Guangzhou, China, in 2006 and 2012, respectively.

He is currently a Professor with the School of Computer Science and Engineering, South China University of Technology, Guangzhou. He has published over 80 papers in international journals and conferences, including over 20 papers in IEEE TRANSACTIONS journals. His current research interests include swarm intelligence algorithms and their applications on cloud computing, operations research, and software engineering.

Dr. Chen was a recipient of the IEEE Computational Intelligence Society Outstanding Dissertation Award in 2016 for his Ph.D. thesis and the National Science Fund for Excellent Young Scholars in 2016.

Jun Zhang (M'02–SM'08–F'17) received the Ph.D. degree in electronic engineering from the City University of Hong Kong, Hong Kong, in 2002.

He is currently a Professor with the University of Victoria, Melbourne, VIC, Australia. His current research interests include computational intelligence, cloud computing, data mining, and power electronic circuits. He has published over 200 technical papers in the above areas.

Dr. Zhang serves as an Associate Editor for the IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION, IEEE TRANSACTIONS ON CYBERNETICS, and IEEE TRANSACTIONS ON INDUSTRIAL ELECTRONICS.